Assuring Fair Allocation of Excess Bandwidth in Reservation Based Core-Stateless Networks

Avadora Dumitrescu, and Jarmo Harju
Tampere University of Technology, Institute of Communication Engineering
P.O. Box 553, FIN-33101, Tampere, Finland
E-mail: avadora.dumitrescu@tut.fi, jarmo.harju@tut.fi

Abstract

Assuring guaranteed services and providing fair bandwidth sharing are much sought characteristics in the current network architecture research. The IntServ approaches achieve those characteristics but only at the expense of using per-flow states and signaling, which are not scalable and difficult to implement. Alternatively, the stateless approaches proposed up to now are focused either on providing guaranteed services similar to those offered by IntServ or on ensuring a reasonably fair bandwidth allocation. In this paper we propose a stateless architecture – the Fair Bandwidth Allocation Control (Fair-BAC) – that ensures both the adherence to the given throughput guarantees and a fair allocation of the remaining bandwidth between the flows that do not have any reservation and the out-of-profile fraction of the flows with reservation. The capabilities of Fair-BAC are demonstrated in a set of simulations performed for various degrees of congestion and using sources with different conformance levels.

1. Introduction

Until recently the network resources-sharing relied almost entirely on end-to-end congestion control mechanisms like TCP. TCP and TCP-friendly sources (i.e. sources whose long-term throughput does not exceed the throughput of a conformant TCP connection under the same conditions [1]) adapt their sending rates according to network condition. If a packet is lost or a timeout event occurs, the TCP sources back-off their transmission rates, permitting the congested links to recover from transient overload.

The new applications (video, IP-telephony, tele-medicine, etc.) that have emerged in recent years do not employ any TCP-friendly congestion and flow control. Instead, many of these new applications, while tolerating some degree of loss, need a continuous minimum bandwidth assurance since a decrease in their bitrate and/or any excess delay could lead to a significant degradation in the user-perceived quality [2]. One solution for these real-time applications is to reserve some bandwidth for the respective flows traversing the network. IntServ architectures [3] (e.g. RSVP) achieve this goal at expense of transmitting and maintaining per-flow state information along the reservation path, which is a complex and not scalable approach. Consequently several core-stateless approaches (i.e. approaches where no per-flow state information is used in the core nodes) were proposed for delivering similar guarantees as the IntServ architectures [4].

It has been argued that, besides reliable bandwidth reservation, fair bandwidth allocation is a necessary requisite for adequate congestion management [5] [6]. Therefore, both per-flow queuing mechanisms [7] [8] [9] and core-stateless [6] [10] [11] [12] approaches were developed for provisioning fair bandwidth distribution.

Per-flow queuing mechanisms operate by maintaining a separate queue for each flow, all the dropping/forwarding decisions being based on the per-flow state. From the core-stateless approaches, CSFQ [6] approximates fair queuing, without maintaining any per-flow state in the core routers, by using state information that is encoded in the packet at the ingress node and is carried, used and updated along the transfer path. Another core-stateless approach, the Rainbow Fair Queuing [10], approximates the statefull Fair Queuing by using a ‘color’ marker representing rate layers of a flow and by discarding the packets in decreasing order of the rate layers to which they belong. In yet another core-stateless approach, Tag-based Unified Fairness (TUF) [11], the labeling takes into consideration the flows’ responsiveness, and the discarding procedure adjusts the drop rates in order to keep the average rates equal.

In this paper we propose a new mechanism – the Fair Bandwidth Allocation Control (Fair-BAC) mechanism – that both guarantees the bandwidth reservations and provides a fair allocation of the excess bandwidth (i.e. the remaining bandwidth capability left after providing all bandwidth reservations) without any per-flow state or per-flow signaling in the core routers. Thus, in case of congestion, each flow receives its own reserved bandwidth (if it has a reservation) and a fair share of the excess bandwidth if it needs.

Section 2 of the paper describes the Fair-BAC mechanism pointing out its particular features and how it is meant to deal with the various network and flow behavior circumstances. Section 3 presents some simulation results and analyses the fulfillment of reserved bandwidth provisioning and the fair distribution of excess bandwidth to several sources with different characteristics under various congestion and reservation conditions. In the end we summarize the capabilities and characteristics of Fair-BAC and outline future work.

2. Fair-Bandwidth Allocation Control - Principles and algorithm description

As usual with QoS architectures, a Fair-BAC compliant domain consists of edge (ingress and egress) nodes and internal (core) nodes. The first principle characterizing Fair-BAC is that that no per-flow state and per-flow signaling is used in the core and egress nodes. The differentiated treatment of the packets, once they entered the Fair-BAC domain, is based on the state of the nodes and on the information carried by the packets themselves. The information inducing packet treatment differentiation is encoded in the packet header and, according to the Dynamic Packet State (DSP) concept [4] [6], it is initialized at ingress and updated along the Fair-BAC path.

The second guiding principle of Fair-BAC is that the reserved bandwidth should be guaranteed and that the traffic with reservation should be isolated from all the effects of congestion as long as it remains in-profile. The application of this principle is based on the assumption that under no circumstances the reservation over a certain network link can exceed the link's capacity and Fair-BAC assumes that the network provider is able to guarantee this.

The third principle is that the remaining capacity, which is left from servicing the in-profile traffic, should be equally divided among the traffic from the sources without reservation and among the out-of-profile fraction of the traffic from the sources with reservation.

Apart from the underlying principles there are several other elements that define the Fair-BAC approach: the information carried by the packets, the congestion state determination, the packet admission decision procedure, the packet buffering and the packet information updating (re-labeling).

The Fair-BAC information carried by each packet includes the bandwidth reservation (*rsv*) and the average bit rate (*abr*) for the flow to which the packet belongs. The *rsv* value, determined according to the flow's Service Level Agreement (SLA), is inserted at the ingress node and does not change while the packet is in the Fair-BAC domain. An *rsv* value of zero is inserted in the packets that belong to traffic without reservation. The *abr* value is initially measured at the ingress node, for example using an exponential moving average (EMA) method, and is subsequently actualized after each hop. In order to save space in the packet header the reserved bandwidth, *rsv*, can be encoded, for example using a logarithmic scale to map rates between $R_{min}=0$ and R_{max} to integer values, and the average bit rate, *abr*, can be encoded as the ratio between *abr* and *rsv*. The granularity of the bandwidth treatment differentiation offered by Fair-BAC obviously depends on the particular encoding and on the number of bits used to encode the bandwidth information into the packet header.

Besides the information carried by the packet, the treatment of the packet in a network node also takes into account the node's internal congestion state. In order to determine the internal congestion state and to evaluate how much of the load for any given link is caused by in-profile traffic and how much by out-of-profile traffic and by the traffic without reservation, two aggregate traffic values are measured in each node: the total traffic aggregate (*Agtf*) and the aggregated in-profile traffic from the sources with reservation (*EffRsvtf*). These two traffic aggregates can be measured using exponential moving average (EMA) or other averaging procedures. In our implementation we used EMA:

$$X_i = \left[1-\exp\left(-\left(t_i - t_{i-1}\right)/W\right)\right]\frac{l_i}{\left(t_i - t_{i-1}\right)} + \exp\left(-\left(t_i - t_{i-1}\right)/W\right)\cdot X_{i-1} \qquad (1)$$

where W is a constant corresponding to the averaging window extension and t_i are the arrival times taken for all incoming packets of a given link. When (1) is used for calculating $Agtf_i$, l_i are the lengths taken for all incoming packets while, when (1) is used for computing $EffRsvtf_i$, l_i are the in-profile lengths of the packets belonging to flows with reservation (i.e. l_i is equal to the packet length when the arriving packet has a reservation greater than *abr*, is equal to *packet length * rsv / abr* when the arriving packet has a reservation smaller than *abr* and is equal to 0 when the arriving packet has no reservation).

Regarding the congestion state we assert that it is not the buffer occupancy that defines congestion but rather the derivative of the buffer occupancy – i.e. congestion is associated with the buffer being filled up while out-of-congestion is associated with the buffer being emptied. Since in this paper we do not discuss details of the buffering mechanisms, the congestion state is identified by the comparison between the aggregated traffic associated with a link (*Agtf*) and the link capacity (*C*). If $Agtf>C$ the link is considered congested (as its associated buffer space is filling up) while if $Agtf<C$ the link is considered not congested (as its associated buffer space is emptying).

Under congestion (i.e. when $Agtf>C$) all in-profile packets - i.e. packets from in-profile sources (sources with $rsv>0$ and $abr<rsv$) and packets corresponding to the in-profile fraction of the out-of-profile sources (sources with $rsv>0$ and $abr>rsv$) are accepted into the buffers and subsequently forwarded. The remaining capacity, C-*EffRsvtf* (where *EffRsvtf* includes also the in-profile fraction of the out-of-profile flows), is distributed according to the max-min fair rate principle, which stipulates that a flow should receive its required bandwidth up to a fair bandwidth share (*fr*) value ensuring that each user's throughput is at least as large as that of all other users which have the same bottleneck [13]. Thus the flows without reservation should be given a bandwidth equal to min{*abr, fr*} and the out-of-profile fraction of the flows surpassing their reservation should be given a bandwidth equal to min{*abr-rsv*,*fr*}, where the fair bandwidth share *fr* is given by the unique solution of the equation:

$$C - EffRsvtf_i = \sum_{j=1,Nf} \min\left\{abr_j - rsv_j,\, fr_i\right\} \qquad (2)$$

The sum in equation (2) is carried out (at successive *i* moments) for the flows without reservation (for which *rsv*=0) and for the out-of-profile flows (for which *abr-rsv* > 0). It should be noted that Fair-BAC does not require solving equation (2) and that we use it just for checking the resulted fair bandwidth allocation.

In order to assure the fair bandwidth distribution under congestion, the out-of-profile packets (i.e. the out-of-profile fraction of the flows that surpass their reservation) and the packets from traffic without reservation should be forwarded with the probability:

$$fp = \min\left\{1, \frac{fr}{abr - rsv}\right\} \qquad (3)$$

with *rsv*=0 for the packets from the flows without reservation. Since there is no way for discriminating between the packets of the out-of-profile traffic that correspond to the in-profile fraction and those that correspond to the out-of-profile fraction an overall forwarding probability should be used for all the packets from out-of-profile flows:

$$fp = \min\left\{1, 1\cdot\frac{rsv}{abr} + \frac{fr}{abr - rsv}\cdot\frac{abr - rsv}{abr}\right\} = \min\left\{1, \frac{rsv + fr}{abr}\right\} \quad (4)$$

Taking into account the inherent bandwidth variation and burstiness, some out-of-profile traffic and traffic without reservation should be accepted beyond their fair share as long as there is buffer space available. This can be done in a fair manner by multiplying the fair rate by a correction factor:

$$corr = \frac{Buffer\_availability}{Buffer\_occupancy}\cdot coef \qquad (5)$$

where *coef* gives the ratio between the buffer occupancy and the buffer availability above which the packet dropping becomes gradually more aggressive in order to keep the buffer from overflowing. When the *Buffer occupancy / Buffer availability* ratio is below the *coef* value the fair rate is increased accordingly, thus accepting some traffic above the fair shares as long as the available buffer space permits. We would like to mention that *Buffer_occupancy* and *Buffer_availability* (=*Buffer_size-Buffer_occupancy*) should also be averaged (for example by using EMA) to smooth-out the sharp variations and to make their variation sensibly slower than that of the estimated fair rate. Otherwise, with the product *fr*Buffer_availability/Buffer_occupancy* estimating the fair share of the excess bandwidth, the congestion and traffic variations will be reflected primarily in buffer status changes and not in *fr* changes.

Thus, taking into account all the previous considerations, the forwarding probability is given by:

$$fp = \left\{ \begin{array}{ll} \min\left\{1, \dfrac{fr}{abr}\cdot\dfrac{Buffer\_availability}{Buffer\_occupancy}\cdot coef\right\} & for\; rsv = 0 \\ 1 & for\; rsv > 0\ \&\ abr \le rsv \\ \min\left\{1, \dfrac{rsv + fr\cdot\dfrac{Buffer\_availability}{Buffer\_occupancy}\cdot coef}{abr}\right\} & for\; rsv > 0\ \&\ abr > rsv \end{array} \right\} \qquad (6)$$

It has to be underlined that the forwarding probability formula (6) ensures that there is no packet loss, not even a small statistical loss, for traffic from in-profile sources. However, there is a statistical loss for the out-of-profile sources, proportional with the out-of-profile fraction of the traffic. Therefore the traffic sources have a strong incentive to keep their traffic below or equal to their *rsv* at all times if they want a guaranteed zero loss in a Fair-BAC domain.

Upon packet acceptance the average bit rate *(abr)* is updated according to the packet forwarding probability:

$$abr\_new = abr\_old * fp \qquad (7)$$

Although (7) holds best when the packets of a flow encounter similar forwarding conditions, the updated *abr* leads to a good overall approximation of the flow's modified bit rate even under non-uniform forwarding conditions as the variation of the flow's bit rate is given by the average of *fp* taken over the flow's packets.

Outside congestion the fair bandwidth share has no sense as the flows are allowed to send at whatever rate. Therefore, outside congestion the variable keeping the estimated value of the fair bandwidth share is assigned a value equal to the maximum of *abr*. This value is updated, as long as the un-congested state persists, by determining the max{*abr*} over a sliding window. When the aggregate traffic surpasses the link capacity (*Agtf*>*C*) it means that at least one flow surpassed its fair share under the given congested conditions. Since, under congestion the fair share of the excess bandwidth has a lower value than max{*abr-rsv*} and therefore lower than max{*abr*} –

otherwise equation (2) would not hold – *fr* must be reduced when entering into congestion.

Instead of calculating the fair bandwidth share under congestion, an instantaneous value is updated using the following formula:

$$fr_{new} = fr_{old} \cdot \left(\frac{C - EffRsvtf_i}{Acctf_i - EffRsvtf_i} \right)^y \qquad (8)$$

where $Acctf_i$ is the averaged traffic effectively accepted for forwarding at moment *i* and *y* is a convergence speed coefficient (the convergence speed of the estimated value for *fr* towards the value given by (2) is increased as *y* is increased above 1 and decreased as *y* is decreased below 1). It should be mentioned that the difference between the time constants of the *EffRsvtf*, *Acctf* and *fr* variations can generate oscillations around the exact solutions. Our current experiences indicate that better convergence and stability is obtained when no averaging is involved in updating the fair share of the excess bandwidth. Furthermore, allowing a rapid variation of the fair share alleviates both the variation effect and the burst effect [11], which diminish the average bandwidth granted to responsive and bursty sources, respectively.

While the accepted traffic (*Acctf*) should be calculated using the same method as for *Agtf* and *EffRsvtf* (EMA in our case) but applied to the traffic accepted into the buffer space, its analytic expression is given by:

$$Acctf - EffRsvtf = \left[Agtf - EffRsvtf \right] \cdot \overleftrightarrow{fp} \qquad (9)$$

with $\overleftrightarrow{fp}$ being the forwarding probability averaged over all packets belonging to flows without reservation or to the out-of-profile fraction of the out-of-profile flows in the estimation interval. It should be mentioned that, similarly with equation (2), equation (9) is not employed in calculations within the Fair-BAC algorithm but is used just for demonstration and evaluation purposes.

When the congestion increases/decreases, in the first instance the traffic effectively accepted for forwarding (*Acctf*) increases/decreases (since no other variable changes initially). Following, the estimated value of the fair bandwidth share is decreased/increased according to (8). Considering that, subsequent to the initial modification, the congestion remains unchanged; the estimated fair bandwidth share is asymptotically stabilized when

$$C - EffRsvtf = \left[Agtf - EffRsvtf \right] \cdot \overleftrightarrow{fp} \qquad (10)$$

which happens when the average forwarding probability $\overleftrightarrow{fp}$ is derived from the fair bandwidth share resulted from (2). Should the stabilization occur at forwarding probabilities corresponding to an estimated *fr* value larger than the fair bandwidth share resulted from (2) then:

$$Acctf - EffRsvtf > C - EffRsvtf \qquad (11)$$

due to the above-fair forwarding probabilities, thus inducing a reduction of the estimated *fr* value through (8). Alternately, should the stabilization occur at forwarding probabilities corresponding to an estimated *fr* value smaller than the fair bandwidth share resulted from (2) then:

$$Acctf - EffRsvtf < C - EffRsvtf \qquad (12)$$

due to the below-fair forwarding probabilities, thus inducing an increase of the estimated *fr* value through (8).

3. Simulation scenarios and results

The simulation scenarios are described in Table I (a, b, c, d) and Table II (a, b, c, d), giving the average bit rates (*abr*), reserved bandwidths (*rsv*), out-of-profile traffic fractions (*abr-rsv*), bandwidth shares that should be fairly allocated on top of the reservation (*max{0,min{abr-rsv,fr}}*) (only *min{abr-rsv,fr}* is mentioned in table headers for space consideration) and fair throughput rates for five constant bit rate and two ON-OFF sources whose traffic is transferred through a core node. The cases a, b, c and d correspond to the four possible combinations of the ON-OFF sources, which lead to different congestion levels and different fair shares of the excess bandwidth.

Table I (Total reservation: 3000 Kbps)

	abr	**rsv**	**abr-rsv**	**min{abr-rsv,fr]**	**throughput**
CBR1	2000	0	2000	2000	2000
CBR2	2000	500	1500	1500	2000
ON-**OFF**1	0	0	0	0	0
CBR3	2000	0	2000	2000	2000
CBR4	2000	2000	0	0	2000
CBR5	2000	0	2000	2000	2000
ON-**OFF**2	0	500	-500	0	0
Total	10000	**3000**	7000	7500	10000
fr		**2000**			

a. Source ON-OFF1 off & source ON-OFF2 off

	abr	**rsv**	**abr-rsv**	**min{abr-rsv,fr]**	**throughput**
CBR1	2000	0	2000	1400	1400
CBR2	2000	500	1500	1400	1900
ON-**OFF**1	0	0	0	0	0
CBR3	2000	0	2000	1400	1400
CBR4	2000	2000	0	0	2000
CBR5	2000	0	2000	1400	1400
ON-OFF2	2000	500	1500	1400	1900
Total	12000	**3000**	9000	7000	10000
fr		**1400**			

b.Source ON-OFF1 off & source ON-OFF2 on

	abr	**rsv**	**abr-rsv**	**min{abr-rsv,fr]**	**throughput**
CBR1	2000	0	2000	1166.67	1166.67
CBR2	2000	500	1500	1166.67	1666.67
ON-OFF1	2000	0	2000	1166.67	1166.67
CBR3	2000	0	2000	1166.67	1166.67
CBR4	2000	2000	0	0	2000
CBR5	2000	0	2000	1166.67	1166.67
ON-OFF2	2000	500	1500	1166.67	1666.67
Total	14000	**3000**	11000	7000.02	10000.02
fr		**1166.67**			

c. Source ON-OFF1 on & source ON-OFF2 on

	abr	rsv	abr-rsv	min{abr-rsv,fr}	throughput
CBR1	2000	0	2000	1500	1500
CBR2	2000	500	1500	1500	2000
ON-OFF1	2000	0	2000	1500	1500
CBR3	2000	0	2000	1500	1500
CBR4	2000	2000	0	0	2000
CBR5	2000	0	2000	1500	1500
ON-**OFF**2	0	500	-500	0	0
Total	12000	**3000**	9000	7500	10000
fr		**1500**			

d. Source ON-OFF1 on & source ON-OFF2 off

Table II (Total reservation: 8000 Kbps)

	abr	rsv	abr-rsv	min{abr-rsv,fr}	throughput
CBR1	2000	0	2000	2000	2000
CBR2	2000	500	1500	1500	2000
ON-**OFF**1	0	1000	-1000	0	0
CBR3	2000	1500	500	500	2000
CBR4	2000	2000	0	0	2000
CBR5	2000	2500	-500	0	2000
ON-**OFF**2	0	500	-500	0	0
Total	10000	**8000**	2000	4000	10000
fr		**2000**			

a. Source ON-OFF1 off & source ON-OFF2 off

	abr	rsv	abr-rsv	min{abr-rsv,fr}	throughput
CBR1	2000	0	2000	1000	1000
CBR2	2000	500	1500	1000	1500
ON-**OFF**1	0	1000	-1000	0	0
CBR3	2000	1500	500	500	2000
CBR4	2000	2000	0	0	2000
CBR5	2000	2500	-500	0	2000
ON-OFF2	2000	500	1500	1000	1500
Total	12000	**8000**	4000	3500	10000
fr		**1000**			

b. Source ON-OFF1 off & source ON-OFF2 on

	abr	rsv	abr-rsv	min{abr-rsv,fr}	throughput
CBR1	2000	0	2000	500	500
CBR2	2000	500	1500	500	1000
ON-OFF1	2000	1000	1000	500	1500
CBR3	2000	1500	500	500	2000
CBR4	2000	2000	0	0	2000
CBR5	2000	2500	-500	0	2000
ON-OFF2	2000	500	1500	500	1000
Total	14000	**8000**	6000	2500	10000
fr		**500**			

c. Source ON-OFF1 on & source ON-OFF2 on

	abr	rsv	abr-rsv	min{abr-rsv,fr}	throughput
CBR1	2000	0	2000	833	833
CBR2	2000	500	1500	833	1333
ON-OFF1	2000	1000	1000	833	1833
CBR3	2000	1500	500	500	2000
CBR4	2000	2000	0	0	2000
CBR5	2000	2500	-500	0	2000
ON-**OFF**2	0	500	-500	0	0
Total	12000	**8000**	4000	2999	9999
fr		**833**			

d. Source ON-OFF1 on & source ON-OFF2 off

Since the link capacity is taken to be C=10000 Kbps, Table I corresponds to a situation where only a limited fraction of the link capacity is reserved while Table II corresponds to a situation where most of the link capacity is reserved. In our simulations we considered a 40s period in which the first ON-OFF source (ON-OFF1) is ON in the interval 20-40s and the second ON-OFF source (ON-OFF2) is ON in the interval 10-30s so that the situations described in Table I and II come up in the simulations in the sequence a (10s), b(10s), c(10s), d(10s).

The output bandwidth variations resulted from simulating the scenarios described in Table I and II are presented in Fig 1 and 2 respectively. For a better readability, the simulated output for the seven sources was distributed on different figures. Fig. 1 a, b and c present the simulated output corresponding to Table 1 scenario for CBR1 and CBR2; CBR3, CBR4 and CBR5; ON-OFF1 and ON-OFF2, respectively, while Fig. 2 a and b present the simulated output corresponding to Table 2 scenario for CBR1, CBR2 and CBR3; CBR4, CBR5, ON-OFF1 and ON-OFF2, respectively. It can be seen that, within the expected variation induced by the statistical nature of the traffic and the probabilistic action of Fair-BAC, the fair throughput rates are obtained for all the sources in all traffic and congestion contexts.

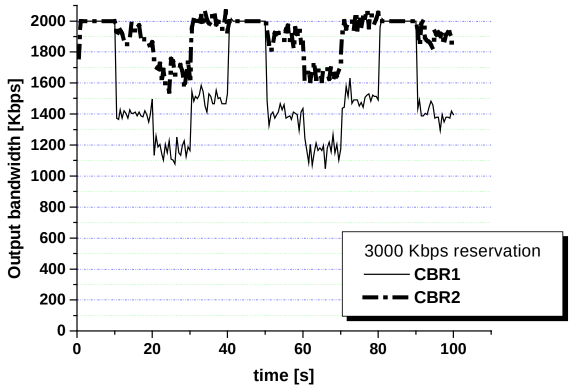

Figure 1a. Simulated output bandwidths for sources CBR1 and CBR2 - corresponding to the Table I scenario

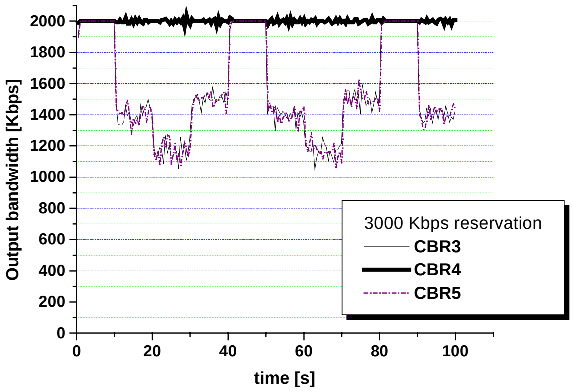

Figure 1b. Simulated output bandwidths for sources CBR3, CBR4 and CBR5 - corresponding to the Table I scenario

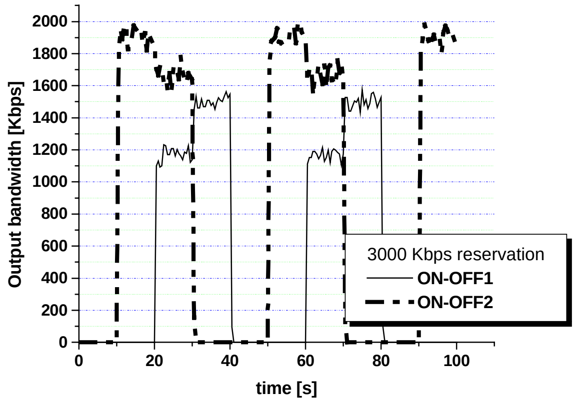

Figure 1c. Simulated output bandwidths for sources ON-OFF1 and ON-OFF2 -corresponding to the Table I scenario

The simulated outputs for CBR3 and CBR5 are barely discernible in Fig. 1b as they are supposed to have similar output bandwidths under the Table I scenario, except from statistical variations.

Some momentary output bandwidth variations come simply from variable buffer occupancy and do not reflect increased output bandwidth or packet loss. For example one can see that, in Fig. 1b, the source CBR4, which should have a constant 2000Kbps throughput, experiences throughput oscillations around the 2000Kbps value – in fact there is no packet loss or output bandwidth increase above 2000Kbps but the variations are due to the variable buffer occupancy that shifts some packet exit times between adjacent measuring intervals. On another hand the buffer occupancy variation does not act when there is no probabilistic forwarding for any of the sources (forwarding probabilities are either 0 or 1), which explains why sometimes the output bandwidths do not have statistical variations.

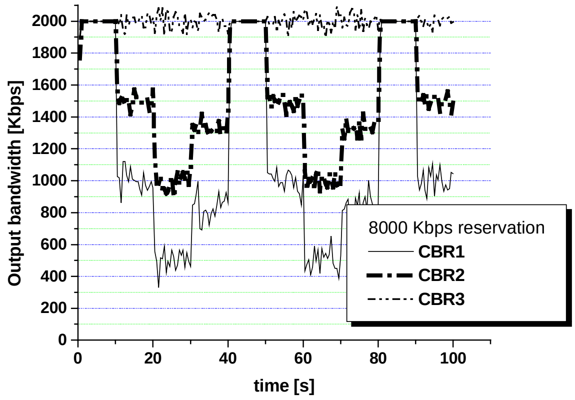

Figure 2a. Simulated output bandwidths for sources CBR1, CBR2 and CBR3 - corresponding to the Table II scenario

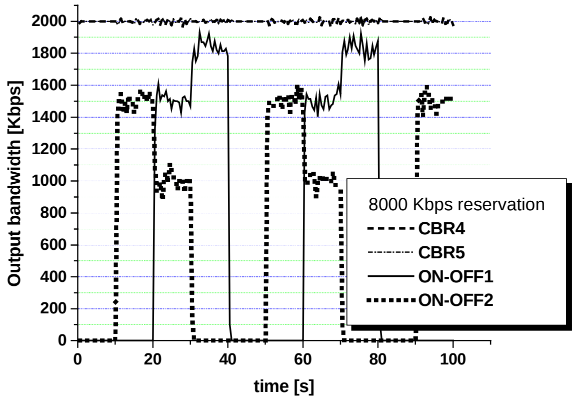

Figure 2b. Simulated output bandwidths for sources CBR4, CBR5, ON-OFF1 and ON-OFF2 – corresponding to the Table II scenario

The simulated outputs for CBR4 and CBR5 are barely discernible in Fig. 2b as they are supposed to have similar output bandwidths (100%) under the Table II scenario, except from statistical variations. The variations of the CBR4 and CBR5 output bandwidth around the 2000 Kbps value are derived from statistical variations of the output moments that translate packet counting between adjacent time intervals – in fact both CBR4 and CBR5 experience zero loss.

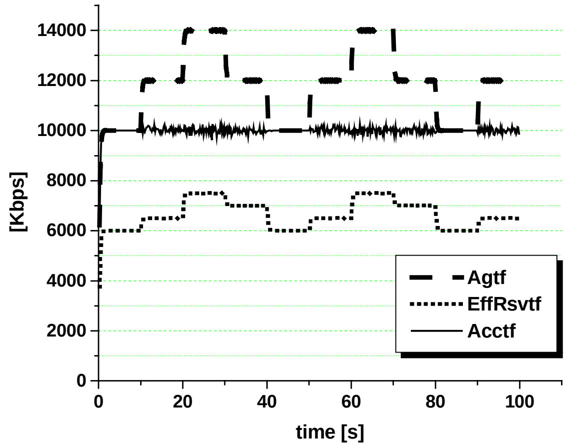

Figure 3. Simulated aggregated traffic, effective traffic with reservation and accepted traffic corresponding to the Table II scenario

Figure 3 presents the simulated aggregated traffic (*Agtf*), effective traffic with reservation (*EffRsvtf*) and accepted traffic (*Acctf*) when a large fraction of the link capacity is reserved (cf. Table II). The supplementary 'noise' in *Acctf* during the congested periods corresponds to statistical variations of the packet output moments and to variations in the buffer occupancy, which, whenever possible, are trying to accommodate the above-capacity input bandwidth and subsequently, when the overload is filling-up the buffer, are discarding more packets in order to reduce buffer occupancy. In the presented simulations the Fair-BAC parameters have been tuned to accommodate only small bursts since, when accommodating bursts that are longer and/or with a higher overload the *Acctf* might experience bigger and more defined oscillations. This situation must be avoided since it might lead to dropping successive packets in groups/clusters.

4. Conclusions and further work

The paper presents a novel core-stateless architecture – the Fair Bandwidth Allocation Control (Fair-BAC) – aimed at ensuring both the delivery of guaranteed services and a fair allocation of the excess bandwidth. A set of simulations using several sources with different characteristics and different conformance levels (in-profile and various out-of-profile degrees) were performed under various degrees of congestion to asses the capabilities of the algorithm. The simulation results have shown that the bandwidth reservations are guaranteed by Fair-BAC and that in-profile traffic is isolated from congestion effects. Also clear reservation-dependent and traffic-context-dependent differentiation of the bandwidth allocation was demonstrated. The fair distribution of the excess bandwidth to the out-of-profile flows and to the flows without reservation is achieved relatively fast after changes in congestion and traffic context. Moreover, the observed variations of the effectively achieved bandwidth distributions with respect to the theoretically fair ones are within the range expected from the statistical nature of the flows and of the Fair-BAC mechanism.

Further work is required to analyze the effect of source burstiness and source responsivity on the fair bandwidth allocation provided by Fair-BAC. Also, the effectiveness of various Fair-BAC elements (like the use of the *Buffer_availability/Buffer_occupancy*coef*) in different contexts, the optimization of the Fair-BAC parameters (*W*, *coef*, *y*) and the optimization of the averaging procedures depending on source behavior, traffic characteristics and congestion patterns require further study. Supplementary, while the described simulations may suffice for the scope of demonstrating the operation of Fair-BAC, more simulations should be used to analyze Fair-BAC's performances (convergence, response speed, stability) and study the optimization of its parameters in a realistic traffic and network context.

Regarding the treatment of maliciously behaving sources our present experiences indicate that short-lived opportunistic flows can exploit to some extent the Fair-BAC provisions meant to accommodate burstiness but this is done on short time intervals (when Fair-BAC is tuned to accommodate only small bursts) and mostly at the expense of temporarily increasing the resource utilization and much less at the expense of other ill-behaved flows or flows without reservation. We have to study further the parameter setup that might ensure both large burst accommodation and prevent malicious short-lived sources from taking advantage from the Fair-BAC tuning.

5. References

1. S. Floyd , K. Fall, "Promoting the use of end-to-end congestion control in the internet", IEEE/ACM Transactions on Networking, vol7, no.4, August 1999, pp. 458-472.
2. V. Servis, "Measuring voice quality over VoIP networks", The TOLLY Group, Dec. 2001.
3. P. White, J. Crowcroft, "Integrated Services in the Internet: state of the art", Proceedings of the IEEE, Volume: 85, Issue:3, April 1997, pp. 277-290.
4. I. Stoica and H.Zhang, "Providing guaranteed services without per flow management", Proceedings of SIGCOMM'99, Boston, MA, September 1999, pp. 81-94.
5. S. Shenker, "Making greed work in networks: A game theoretical analysis of switch service disciplines", Proc. ACM SIGCOMM, London, Aug. 1994, pp. 47-57.
6. I. Stoica, S. Shenker, H. Zhang, "Core-stateless fair queuing : a scalable architecture to approximate fair bandwidth allocations in high speed networks", IEEE/ACM Transactions on Networking, Volume: 11, Issue: 1, February 2003, pp. 33-46.
7. A. Demers, S. Keshav, S. Shenker, "Analysis and simulation of a fair queuing algorithm", in Proceedings of ACM SIGCOMM'89, 1989, pp. 3-12.
8. A. Parekh, R. Gallager, "A generalized processor sharing approach to flow control -The single node case", IEEE/ACM Transactions on Networking, vol.1, June 1993, pp. 344-357.
9. M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round robin", Proceedings ACM SIGCOMM, Boston, MA, Sept. 1999, pp. 135-146
10. Z. Cao, Z. Wang, E. Zegura, "Rainbow fair queuing: Fair bandwidth sharing without per-flow state", Proceedings of INFOCOMM'99, Tel-Aviv, Israel, Mar 2000, pp.922-931.
11. A. Clerget, W. Dabbous, "TUF: tag-based unified fairness", Proceedings of INFOCOM 2001, Anchorage, AK, April 2001, pp. 498-507.
12. H. Zhu, A. Sang, S.-Q- Li, "Weighted fair bandwidth sharing using SCALE technique", Computer Communications, 24, 2001, pp. 51-63.
13. J.M. Jaffe, " Bottleneck Flow Control", IEEE Trans. Commun., vol. 29, no.7, July 1981, pp. 954-962.